EXHIBIT "Y"

UNITED STATES DISTRICT COURT
EASTERN DISTRICT OF NEW YORK

NABIL N. GHALY,

Plaintiff,

-v.-

HASBRO, INC.,

Defendant.

MEMORANDUM OF DECISION AND ORDER
97-CV-7037 (DRH) (VVP)

NABIL N. GHALY,

Plaintiff,

-v.-

TIGER ELECTRONICS, INC.; TIGER ELECTRONICS, LTD. as true party in interest as successor to Tiger Electronics, Inc.; LION HOLDINGS, INC.; ROBERT DUNN GLICK, as Trustee for liquidating trust for Lion Holdings, Inc.; ROBERT DUNN GLICK, as Trustee for the Rissman Family 1997 Trust; OWEN RANDALL RISSMAN; TIGER ELECTRONICS, LTD.; and HASBRO, INC.,

Defendants.

98-CV-5239 (DRH)(VVP)

Appearances:

For the Plaintiff
Nabil N. Ghaly, *pro se*
14 Longwood Dr.
South Huntington, N.Y. 11746

For Defendants
Marshall, O'Toole, Gerstein, Murray & Borun
6300 Sears Tower
233 South Wacker Drive
Chicago, IL 60606
By: Edward M. O'Toole, Madeline H. Devereux, and Anthony G. Sitko

Nims, Howes, Collison, Hansen & Lackert
605 Third Ave., Suite 3500
New York, N.Y. 10156
By: William Robert Hansen

HURLEY, District Judge:

The Plaintiff is the holder of United States Patent No. 5,286,037 ("the '037 patent"), issued on February 14, 1994, for an "electronic hand-held logic game." In this action, he sues the Defendants, alleging that several of their products infringed on the '037 patent. Presently before the Court is the Defendants' motion for summary judgment on the issue of claim construction, as required by *Markman v. Westview Instruments, Inc.*, 517 U.S. 370, 372 (1996) ("construction of a patent, including terms of art within its claim, is exclusively within the province of the court."), and the Plaintiff's motion to strike a portion of the Defendants' brief as irrelevant and prejudicial.

BACKGROUND

In general, the Plaintiff's device is a hand-held electronic game consisting primarily of a playfield composed of a matrix of buttons, each illuminated in one of several colors. (As described in the perferred embodiment, the playfield is a four-by-four square, consisting of a total of 16 buttons.) A player pressing one of the buttons will cause that button, as well as possibly one or more additional buttons on the playfield, to change to a different color. The number and position of the buttons changing color, as well as the color(s) they change to, are determined by various algorithms programmed into the game. The player's goal is to manipulate the various buttons in such a manner as to cause all of the buttons on the playfield to display the same color.

More specifically, the Plaintiff distinguishes his device from previous hand-held logic puzzle games by the particular methodin which the game calculates the initial starting conditions and the effect each button has on the other buttons. Put as simply as possible, the Plaintiff's device initiates a game by randomly assigning values to each of eight locations along the left and bottom edges of the matrix (the "transmitters"), and to each of eight locations in the top and right edges (the "receivers"). The specification and diagrams refer to these values as "operating codes." Next, the device queries the status of each button on the playfield as either "on" or "off." The device then establishes an initial set of routes through the playfield, connecting each transmitter location along the bottom and left of the playfield with a receiver location along the top and right. The actual path of the route is determined by a predefined "routing square" which perscribes one of two paths through the square based on the status of the button in that square– that is, a route that passes through a particular button which is in the "on" position will be re-routed when that button is in the "off" position.

Having established the routes, the device next assigns starting colors to each of the receiver locations along the top and right edges of the playfield. The assignment of the colors is made by passing each receiver value and the corresponding transmitter value at the other end of its route through specified Boolean logic[1] functions. The result of the Boolean functions yields a three digit binary number referred to as a "color code," one of which is assigned to each of the

---

[1]Boolean logic is used to perform operations on binary numbers, such as those used in the operating codes. In general, Boolean logic involves functions (*e.g.* "the two values are the same" or "one, and only one, of the two values is a zero") which compare two binary values and return a single digit result indicating that the conditions required by the function are met ("true") or not met ("false").

eight receiver locations on the top and right edges of the playfield. The device then creates a new set of routes backwards, from the receiver locations to the transmitter locations, again based on the routing square and the status of the buttons that each route passes through along the way.

As a result of these steps, each button is assigned four routes: a route from a transmitter to a receiver when the button is in the "on" position; a different route from a transmitter to a receiver when the button is in the "off" position; a route from a receiver to a transmitter when the button is in the "on" position; and a route from a receiver to a transmitter when the button is in the "off" position. The latter two routes, from receiver to transmitter, also carry the color codes assigned to the receiver locations they originate at, and thus, each button carries two assigned color codes.

At this point, the device is ready to begin play. The device illuminates each button in the proper color based on the color codes routed through it and its current status of "on" or "off." The device then checks to see if all buttons are the same color, and, if so, diverts to a new process described below. If all the buttons are not currently the same color, the unit plays a sound to alert the player that it is ready for the player's next move, and waits. When the player presses one of the buttons, he changes that button's status from "on" to "off" or vise-versa. The device then proceeds to re-generate a full set of new routes from the transmitter locations to the receiver locations, re-assign the colors to the receiver locations, and re-generate return routes for the color codes, and display the proper colors for each button, all in the manner described above.

Obviously, because the player has changed the status of one of the buttons on the playfield, the new route from some of the transmitters to some of the receivers will be different from those that existed before the button was pressed. For example, while transmitter 3 may

have previously been matched to receiver 6, the re-routing of the playfield might now result in transmitter 3 connecting to receiver 1. This change in the correspondence between transmitter and receiver values might result in a different color code being generated and assigned to the receiver when the transmitter and receiver values are compared using the Boolean functions. Moreover, the change might also alter the routes back through the device that carry the color codes from the receivers to the transmitters. As a result, each press of a button changes that button's color, and may also affect one or more colors of other buttons on any of the changed routes may change as well. An additional consequence of the unique routing approach by the Plaintiff is that a button which once caused its neighbors to the right and left to change color might, as a result of newly assigned routes from buttons pressed elsewhere, later cause buttons on only its top and bottom to change color.

If all of the buttons display the same color, the device plays a tune associated with the color shown and causes all of the buttons to flash briefly and then display random colors while the tune plays. The device notes the color that the player has solved, and then checks to see if the player has solved all of the colors involved in the game. If not, the game returns to regular play mode, and the player continues to attempt to solve the remaining colors. Once all of the colors are solved, the device plays a tune accompanied by flashing buttons, and the game ends.

Primarily at issue are Claims 1 and 23 in the '037 patent. Claim 1 consists of 9 elements, and reads as follows:

1. An electronic game device comprising:

a. a housing for the device;

b. means for generating a plurality of codes hereinafter referred to as operating codes;

c. plurality of entry control means;

d. plurality of routing means defining a respective plurality of playing positions on the surface of said housing, each of said routing means being actuable by said entry control means to route said operating codes within the device;

e. means to generate a plurality of codes, hereinafter referred to as color codes, from said plurality of operating codes;

f. plurality of multi-color light emitting means;

g. means to route said color codes to said light emitting means in accordance with the determination of said routing means;

h. means to decode said plurality of color codes and activate said plurality of multi-color light emitting means;

i. means for varying the level of difficulty of any particular game; and

j. Sensorially perceptible indicating means responsive to said entry control means for generating a first sensorially perceptible indication corresponding to each activation of the entry control means, a plurality of sensorially perceptible and distinct indication of each of which is corresponding to each of a plurality of predetermined colors being displayed at al (sic) multi-color light emitting means and a sensorially perceptible indication corresponding to the successful completion of a game.

Claim 23 is similar to Claim 1, with the primary exception being element e of Claim 23, which replaces element f of Claim 1, and reads "means to pictorially represent a plurality of images

wherein each of said plurality of playing positions is indicated to provide a plurality of display positions, each of said display positions is used to indicate any of said plurality of images." In addition, Claim 23 replaces Claim 1's reference to "color codes" with "display codes." The parties dispute the construction to be given to nearly every element of Claim 1, and to element e of Claim 23.

A. Governing law

The threshold inquiry for construing a patent claim as a matter of law is to "look to the words of the claims themselves, both asserted and nonasserted, to define the scope of the patented invention." *Vitronics v. Conceptronic, Inc.*, 90 F.3d 1576, 1582-83 (Fed. Cir.1996). Terms in a claim are "generally given their ordinary and customary meaning" unless the patentee has clearly stated special definitions in the patent specification or file history. *Id.* Second, it is necessary to review the patent specification, which is usually "dispositive; it is the single best guide to the meaning of a disputed term." *Id.* "Third, the court may also consider the prosecution history of the patent, if in evidence." *Id.* Unless the intrinsic evidence is "genuinely ambiguous," it is improper to rely on extrinsic evidence, such as expert and inventor testimony, in construing the claim. *Interactive Gift Express, Inc. v. Compuserve Inc.*, 256 F.3d 1323, 1332 (Fed. Cir.2001); *Robotic Vision Sys., Inc. v. View Eng'g, Inc.*, 189 F.3d 1370, 1375 (Fed. Cir.1999) ("Often, the intrinsic evidence alone will resolve any ambiguity in a disputed claim term, and, in such instances reliance on extrinsic evidence is improper.")

Additionally, "a patent is not to be limited to the preferred embodiments shown in the specification." *Ziegler v. Phillips Petroleum Company*, 483 F.2d 858, 879 (5th Cir.1973), *cert. denied*, 414 U.S. 1079 (1973), *citing Continental Paper Bag Co. v. Eastern Paper Bag Co.*, 210

U.S. 405 (1908). However, the designation of a particular embodiment as "preferred" does not of itself broaden the claims beyond their support in the specification. *Wang Labs., Inc. v. America Online, Inc.*, 197 F.3d 1377, 1383 (Fed. Cir.1999). While a claim must be read in light of the specifications, limitations from the specification may not be read into the claims. *Bell Atlantic Network Service, Inc. v. Covad Communications, Inc.*, 262 F.3d 1258, 1270 (Fed. Cir. 2001)

Many of the elements of the Plaintiff's claims are written in "means plus function" format, in that they describe "a means or step for performing a specified function without the recital of structure, material, or acts in support thereof." 35 U.S.C. § 112, ¶ 6. Construction of these claims requires the court to identify the function being performed in the claim and determine what structures have been disclosed in the specification that correspond to the means for performing that function. *Kemco Sales, Inc. v. Control Papers Co.*, Inc., 208 F.3d 1352, 1361 (Fed. Cir.2000). Such claims "shall be construed to cover the corresponding structure, material, or acts described in the specification." 35 U.S.C. § 112, ¶ 6.

B. As to Claim 1, Element b- "means for generating a plurality of codes"

Claim 1, Element b reads "means for generating a plurality of codes hereinafter referred to as operating codes." The Plaintiff contends that this element should be construed as "structures that generates a first set of codes ('operating codes') for the purpose of generating a second set of codes ('color codes')," namely, "any structure connected to, or incorporated in a hard-wired control logic, and which generates electrical signals or codes"; "a control logic on a processor that generates a plurality of codes or signals in binary form. . ."; or "a control logic executed on a processor that includes a data section containing a plurality of codes or signals in binary form. . . ." The Defendant contends that this element should be construed as "control

extent to which the routing squares are used. The proper construction given to Element d as a whole, then, is "control logic executed on a processor that defines paths for operating codes through the switches on the playfield by means of a routing square which prescribes predefined routes based on the position of said switches."

D. Claim 1, Element e- "means to generate a plurality of . . . color codes"

Claim 1, Element e states "means to generate a plurality of codes, hereinafter referred to as color codes, from said plurality of operating codes." The Plaintiff contends that this element should be construed as any of "a control logic executed on a processor that performs an appropriate boolean function on pairs of operating codes to generate color codes"; "a control logic executed on a processor that determines color codes associated with pairs of operating codes by looking up data stored in a data section of ROM"; or "a plurality of boolean logic devices incorporated in a hard-wired control logic." The Defendant contends that the element should be construed as "control logic executed on a processor that performs two mathematical functions (namely an inclusive OR and an XOR function) on the transmitter op-code and receiver op-code pairs at each of the locations on two edges of the field of play to produce color codes at each of these locations at the positions at the top and right edges of the field of play."

The specification describes the method of generating color codes at 6:18-47. In particular, it states that "the central processing unit . . . generates a color code at each opcode receiver." As described above, the color codes are generated at each of the receiver locations by applying specified boolean logic functions to the pairs of connected operating codes, one digit at a time. 6:34-42. The description of the preferred embodiment in the specification explains that the leftmost digits in the operating codes are subjected to what the specification describes as an

"INCLUSIVE OR" function.[3] 6:34-40. The two remaining digits are subjected to the "EXCLUSIVE OR" function. 6:40-43. However, in a mathematical explanation of the logic to be used for any size playfield, the Plaintiff does not specifically describe the boolean function to be used, supplying only a variable to represent the function to be applied to the operating codes. 12:57-59. This suggests that other embodiments of the game might require different boolean functions to generate proper color codes.

Because element e is written in means plus function language, the Court must identify those structures by which the device generates color codes. The patent clearly states that "to generate the color code at receiver '1,' the central processing unit" executes the appropriate boolean functions. 6:34-35. Therefore, the structures that appear to be used to generate the color codes are the central processing unit and the boolean functions. No other structures appear to be involved in the generation of the color codes.

Accordingly, the proper construction of element e is "control logic executed on a processor that performs predefined boolean functions on the operating code pairs."

F. Claim 1, Element f - "plurality of multi-color light emitting means"

Element f claims a "plurality of multi-color light emitting means." The Plaintiff contends that this should be construed as either "any display or structure that provides visual representation of the outputs (color codes), of an electronic puzzle, in two or more colors that

---

[3]There is a substantial dispute between the parties over the Plaintiff's reference to the "INCLUSIVE OR" function. The Plaintiff admits that while the specification refers to the "INCLUSIVE OR" function, he intended to describe the operation of the "EXCLUSIVE NOR" function. Figure 23, which shows a matrix by which color codes are assigned, appears to actually apply the "EXCLUSIVE NOR" function.

M. The Plaintiff's motion to strike

The Plaintiff separately moves to strike Section IV of the Defendants' brief on the grounds that it improperly interjects consideration of the accused devices into the claim construction process. Section IV of the Defendants' brief describes the lineage of the accused devices as deriving from a hand-held game called "Merlin," created by Parker Brothers in the late 1970s. One of Merlin's operations was a game entitled "Magic Square," in which players attempted to illuminate a specific pattern of lights on a 3 x 3 playfield, where each press of a button changed the status of that button and several others on the playfield in a predetermined fashion. The Defendants go on to explain the basic operation of "Merlin-type devices," as well as a summary of how the accused devices operate. The Plaintiff alleges that such information is irrelevant to the claims construction process and should be stricken as immaterial and prejudicial.

As a general matter, patent claims are construed objectively, without respect to the accused devices. *SRI Int'l v. Matsushita Electric Corp. of America*, 775 F.2d 1107, 1118 (Fed. Cir.1985). However, the court is only required to construe those claims which are actually in dispute. *Vivid Technologies, Inc. v. American Science & Engineering,* Inc., 200 F.3d 795, 803 (Fed. Cir. 1999). The claims in dispute are necessarily framed by the intersections of the patented and accused products, and thus, the court's awareness of the accused products helps to limit the court's consideration to only those claims that are genuinely in need of construction. *See Scripps Clinic & Research Foundation v. Genetech, Ltd.*, 927 F.2d 1565, 1580 (Fed. Cir. 1991) ("Of course the particular accused product (or process) is kept in mind, for it is efficient to focus on the construction of only the disputed elements or limitations of the claims.").

In this regard, the Court has considered the Defendants' recitation of Merlin and its progeny solely as background information, particularly with regard to the distinction between previous generations of "hard-wired" games, in which a particular button has a predefined and unchanging effect on other buttons, and the Plaintiff's innovation, which generates unique relations between buttons both between different games and indeed, within each individual game. Consideration of how Merlin and the acused devices operate helps to constrain the claims to be construed here to only those claims that are or may be present in one or more of the accused devices. The Court has given no consideration whatsoever to the contents of Section IV of the Defendants' brief for purposes of construing the disputed claims themselves, and has relied entirely on the contents of the '037 patent for that purpose. Accordingly, the Plaintiff's motion to strike is denied.

CONCLUSION

For the foregoing reasons, the Defendants' motion to for summary judgment is GRANTED insofar as the Court has construed the disputed claims in the '037 patent as a matter of law. These constructions are set forth herein. The Court expressly declines to make any finding of actual infringement at this time. Because actual infringement is a question of fact, the Court anticipates that a trial will be necessary, unless either party makes a letter request for leave to move for summary judgment on the issue of actual infringement within 45 days of the date of this order.

The Plaintiff's motion to strike Section IV of the Defendants' brief is DENIED.

SO ORDERED.

Dated: Central Islip, N.Y.
March 5, 2002

Denis R. Hurley,
United States District t Judge

Color Switching Games

Gregory P. Tollisen* AND Tamás Lengyel*
Mathematics Department, Occidental College
1600 Campus Road, Los Angeles, CA 90041

ABSTRACT. In this paper the authors study one- and two-dimensional color switching problems by applying methods ranging from linear algebra to parity arguments, invariants, and generating functions. The variety of techniques offers different advantages for addressing the existence and uniqueness of minimal solutions, their characterizations, and lower bounds on their lengths. Useful examples for reducing problems to easier ones and for choosing tools based on simplicity or generality are presented. A novel application of generating functions provides a unifying treatment of all aspects of the problems considered.

1 Introduction

The motivation for this paper is to illustrate several approaches to solving color switching games. At its most general, a color switching game is a one-person game played on a set of locations (e.g., squares on a chessboard or the vertices of a graph) each colored black or white, which we call the "coloring pattern." The player makes a move by changing the color of the location of her choice. However, each location is associated with a set of neighbors in a predetermined way. By changing the color of one location, the colors in the associated locations also are changed automatically. Given initial and final coloring patterns, the game is won if the player successfully "moves" from the one pattern to the other. We begin by focusing on the following two problems found in [9] (Exercises 3.4.25 and 3.4.26, p.117) in a slightly different form.

Problem 1. Consider a row of $2n$ squares colored alternately black and white. A legal move consists of choosing any set of contiguous squares (one or more squares with no gaps allowed), and inverting their colors. What is the minimum number of moves required to make the entire row of one color?

Problem 2. Answer the same question as above, except now we start with a $2m \times 2n$ checkerboard and a legal move consists of choosing any subrectangle and inverting its colors.

Clearly, in Problem 1, n moves will work, for we can invert the 1st square,

*e-mail: tollisen@oxy.edu and lengyel@oxy.edu.

then the 3rd square, etc. In Problem 2 we can switch all even numbered rows and then all odd numbered columns and thus solve the problem in $m + n$ moves. Not only will we show that these are best possible but we will completely characterize all optimal solutions to Problem 2.

We also generalize these problems to linear and rectangular boards with arbitrary initial colorings and moves of both fixed and arbitrary sizes. In each case, the game is won by attaining one color, say white, throughout the entire board. We are interested in the following questions: Can we win the game? If so, what is the minimum size of a winning set of moves? Is a minimal winning set (a winning set of minimum length) unique? Whether or not a minimal winning set is unique, is there a simple description of a minimal winning set, or if not, is there a simple algorithm that will generate a minimal winning set?

The discussion revolves around various uses of linear algebraic tools, parity arguments, invariants, and generating functions in connection with modular arithmetic. We also illustrate their comparative advantages and inherent limitations. Generating function based techniques have been used for analyzing problems on checkerboard tilings [3], checker-jumping [1] and sumset of multisets [3], among others. Typically, generating functions are used to restate and solve a given problem entirely within an algebraic environment. In contrast, we use generating functions as an intermediate step in exchanging a given game for an equivalent board game whose solutions are almost transparent.

We need some terminology. The game board consists of labeled squares. We use natural labeling: in the linear case, the squares can be labeled from 1 to $2n$, and for rectangular boards we label the squares row by row. We call an assignment of colors to the squares a coloring pattern. The block of contiguous squares (in the linear case) or the rectangular block of squares (in the 2-dimensional case) used for inverting colors, we call the switching pattern. We refer to a switching pattern by indicating the position on the board occupied by its leftmost or upper left corner square, respectively, and by its size when needed.

For the linear board consisting of $2n$ squares, we have the following four theorems.

Theorem 1 *Suppose the board is colored so that adjacent squares have alternate colors. If the possible moves are switching patterns of arbitrary length, then the game can be won and the minimum number of moves needed is n.*

Theorem 2 *Suppose the board is colored as in Theorem 1, and the possible moves are switching patterns of fixed length r. Then the game can be won if and only if $r|n$, and the minimum number of moves needed is n.*

Theorem 3 *Suppose the board is arbitrarily colored. If the possible moves*

are of arbitrary length, then the game can always be won and the minimum number of moves needed is equal to the number of blocks of adjacent black squares on the board.

Theorem 4 *Suppose the board is arbitrarily colored and the possible moves are switching patterns of fixed length r. Then the game can be won if and only if the parity conditions*

$$\sum_{i \equiv a \bmod r} (c_i - c_{i-1}) \equiv 0 \mod 2$$

are met for all $0 \leq a < r$. (Here $c_{-1} = 0$.) Equivalently, the game can be won if and only if the sums

$$S_a = \sum_{i \equiv a \bmod r} c_i, \quad 0 \leq a < r,$$

all have the same parity.

Note: In Theorems 1 and 3, the obvious solutions are minimal solutions, but generally, there will be other minimal solutions as well. Theorem 1 is proved in Section 3, and Theorem 3 can be easily proved in the same fashion. Theorems 2 and 4 can be proved by a single variable version of the generating function arguments applied in Section 4. In both theorems, the minimum solution is unique and is obtained by the greedy algorithm.

The following theorems concern a game board of size $2m \times 2n$.

Theorem 5 *Suppose the board is checkerboard colored and the moves are switching patterns of arbitrary sizes. Then the game can be won and the minimum number of moves needed is $m + n$. A minimal solution is obtained by switching the colors of the even numbered rows and odd numbered columns. All minimal solutions can be completely characterized by their common structure.*

Theorem 6 *Suppose the board is checkerboard colored and the moves are of fixed size $s \times t$. Then the game can be won if and only if $s|m$ and $t|n$, the minimum number of moves needed is $\frac{2(s+t-1)}{st} \cdot mn$, and the solution is unique.*

Theorem 7 *Suppose the board is arbitrarily colored and the moves are arbitrarily sized. Then the game can always be won, there is generally no unique minimal solution, and the minimum number of moves needed is bounded above by $mn + O(\min(m\sqrt{n}, n\sqrt{m}))$ as $m, n \to \infty$.*

Theorem 8 *Suppose the board is arbitrarily colored and the moves are of fixed size $s \times t$. Then the game can be won if and only if the parity conditions (7) are met. The minimal solution is unique.*

Note: We present two proofs of Theorem 5 in Sections 3 and 4.2. The characterization of the minimal solutions is given in Section 4.2. In Theorems 6 and 8, the greedy algorithm where one goes through the board row by row gives the minimal solution. In Section 4.1 we give the proof for Theorem 8 and a sketch for Theorem 6. We believe that there is no easy way to get a minimal solution in the case of Theorem 7, whose proof is given in Section 4.3.

We discuss the linear algebraic technique in Section 2. In Section 3, we explore parity arguments and generalized invariants to establish lower bounds on the minimum number of moves for alternating and checkerboard colorings. The most general approach is based on various uses of generating functions as explained in Section 4. In the last part of the paper we list further results.

2 Linear algebraic approach

The problems have a flavor similar to that of games like Merlin's magic squares ([6]-[8]). They can be expressed in terms of 0-1 problems, and often they are analyzed and solved by linear algebraic techniques. In these games, it is common to have a simple association between the N squares of the board and the M potential moves. We define two N dimensional vectors, c_1 for the starting and c_2 for the target coloring.. The main idea is that an arbitrarily colored board with N squares can be represented by an N-dimensional 0-1 vector where 1 stands for a black square while 0 for a white one. The potential moves can be described by matrix A of size $N \times M$ with 0-1 entries by setting $a_{ij} = 1$ if using the jth potential move switches the value at position i, and 0 otherwise.

Clearly, every move should be used only once or not at all, and the order of the moves is irrelevant. This is a common feature of additive games with objects that alternate states [8]. We can describe the series of moves by a set of 0-1 coordinates: 1 if the potential move is applied and 0 otherwise. The corresponding vector of actual moves will be denoted by x. Therefore, we can find the solution to any problem getting from coloring c_1 to c_2 by solving

$$c_2 \equiv c_1 + Ax \bmod 2. \tag{1}$$

Note that the same solution will take us from c_2 to c_1. In many situations we have $M = N$. For instance, for Merlin's game we get $M = N = 9$. In this case, the necessary and sufficient condition for solving the problem of getting from any c_1 to any c_2 by applying matrix A is that A be invertible mod 2.

For example, in Theorem 8, for the board of size $2m \times 2n$ with fixed rectangular switching pattern of size $s \times t$, we note that the problem can be reduced to a truncated version of the board consisting of the upper left

hand portion of size $N = (2m-(s-1)) \times (2n-(t-1))$. For the moment we only care about how the moves affect this portion of the board. By labeling the squares row by row and defining the jth potential move as the switching pattern positioned at the jth square, we get a lower triangular matrix with ones on the diagonal. This fact guarantees a unique solution on the truncated board which uniquely extends to the original. If it matches the original coloring then we have the solution. The truncated version forces conditions on the entire board captured by parity restrictions which we will derive in Section 4. Note that the number of ones in the solution vector x gives us the number of moves used.

Particular solutions to the games and sometimes even their uniqueness can be derived by this technique, e.g., in the cases of Theorems 6 and 8, although without much insight into the size of the minimal solution. One limitation to the linear algebraic restatement is that the 2-dimensional structure of the original problem is reduced to a linear one which does not reflect adjacency. Another concern is that while setting the problem in matrix form seems fairly easy when there is a unique move associated with every position, it becomes more complicated when M is larger than N.

We note that Lovász [5, Exercise 5.17] discusses color switching games in graphs with N vertices. In this case A can be set to be a symmetric 0-1 matrix of size $N \times N$ with all ones in its diagonal. It is proven that there is always a solution to equation (1) with c_1 and c_2 being the all zero and all one vectors, respectively (cf. [4] and [2]).

3 Parity arguments and invariants

If we are asking questions regarding existence or non-existence, parity and invariance may first come to mind. The conditions alluded to in the previous example with fixed size switching patterns are equivalent to a set of parity conditions. Later on we will see how the parity conditions fall out as a by-product of the generating function approach. For switching patterns of arbitrary sizes, parity arguments do not seem to help. Of course, for these problems existence is not an issue. Even more, a greedy approach provides a direct solution. The question is whether this is best possible in the sense that it requires the minimum number of moves. One might suspect that parity based invariants may not advance our efforts to answer this question. Fortunately, a slight generalization of the concept of invariance comes to the rescue.

We consider an invariant to be an aspect of a given problem–usually a numerical quantity–that does not change, even if many other properties do change (cf. [9]). On the other hand, a pseudo-invariant is a quantity which may or may not change at each step of a problem but when it does change, it does so in a limited way.

For Problem 1, by setting $c_i = 1$ if square i is black and 0 otherwise, we

use the following pseudo-invariant. Let

$$g(c_1, c_2, ..., c_N) = \sum_{i=1}^{N-1} |c_i - c_{i+1}| \tag{2}$$

which changes only by -2, -1, 0, 1, or 2 for any legal move. In Problem 1 we have $N = 2n$ and note that the initial value of g is $2n - 1$. It shows that it takes at least n moves to reach 0.

We now show that Problem 2 reduces to Problem 1, and that the checkerboard coloring of a $2m \times 2n$ board with the upper left hand corner colored black, and with arbitrary switching patterns needs at least $m + n$ moves. The proof follows from Problem 1 once we find a "linearizable obstacle" that is long enough. This will also work for many "mutilated boards" derived from a checkerboard. A sufficient condition for this lower bound is that the remaining board has a particular subset of squares called a "NW-SE snake." We call a sequence of contiguous squares a "NW-SE snake" if it starts and ends at the NW and SE corners respectively, and always goes from North to South and from West to East. If a board has a "NW-SE snake" then reduction to Problem 1 works. As a matter of fact, we have a structure with $N = 2n + 2m - 1$ squares. Notice that any rectangular color switching pattern will only change neighboring squares of the snake. Function g has the initial value $2n + 2m - 2$ and the game ends with the value 0, so while it initially appears that we need $n + m - 1$ moves we actually need one more, for a black end requires a color switch that decreases g by at most 1. A similar argument can be used to obtain a lower bound on the minimum number of moves for an arbitrarily colored mutilated board by finding a NW-SE snake of maximum length which has a black square on at least one end.

4 Generating functions

In this section, our strategy is to use generating functions to translate our games into an algebraic description. We develop two equations (5) and (6) for arbitrary and fixed size switching patterns, respectively. For arbitrarily sized switching patterns, we reinterpret the result as another similar game through which we may gain some insight into the first game. All theorems of this paper can be proven by this technique.

A coloring pattern (or coloring) of a $2m \times 2n$ game board is a $(0, 1)$ matrix $C = [c_{ij}]_{\substack{0 \le i < 2m \\ 0 \le j < 2n}}$ where $c_{ij} = 1$ if and only if the square at position (i, j) is colored black. Its generating function is

$$\Phi_C(x, y) = \sum c_{ij} x^i y^j.$$

(For convenience, we specify that $c_{ij} = 0$ outside of the game board, unless otherwise indicated.) For example, the checkerboard coloring with the upper left-hand square colored black has generating function

$$\Phi_C(x,y) = \sum_{\substack{i<2m, j<2n \\ i+j \text{ even}}} x^i y^j = (1+xy)\left(\frac{1-x^{2m}}{1-x^2}\right)\left(\frac{1-y^{2n}}{1-y^2}\right). \tag{3}$$

Solving a coloring $C = [c_{ij}]$, i.e., finding a sequence of rectangular moves that takes us between C and the all white board is equivalent to finding values for the coefficients m_{klst} that solve the polynomial equation

$$\sum m_{klst} x^k y^l \left(\frac{1-x^s}{1-x}\right)\left(\frac{1-y^t}{1-y}\right) \equiv \sum c_{ij} x^i y^j \tag{4}$$

where the indices in the sum on the left are restricted to $0 \le k < 2m, 0 \le l < 2n, 0 < s \le 2m-k$ and $0 < t \le 2m-l$. (This congruence and all others are understood to be modulus 2, unless otherwise indicated. As a consequence, we can freely change the sign of any additive terms.) Then $m_{klst} = 1$ if and only if the move of size $s \times t$ with upper left corner occupying position (k,l) on the game board is included in the winning sequence.

If we multiply both sides of equation (4) by $(1-x)(1-y)$ and collect equal powers of x and y on the right, we get the equivalent equation

$$\sum m_{klst} x^k y^l (1-x^s)(1-y^t) \equiv \sum_{\substack{0 \le i \le 2m \\ 0 \le j \le 2n}} x^i y^j \left[c_{ij} - c_{i,j-1} - c_{i-1,j} + c_{i-1,j-1}\right], \tag{5}$$

which will lead to a new game discussed in Sections 4.2 and 4.3.

Moreover, Section 4.1 will employ the result that, if the size of the moves permitted in the original game is fixed, then $m_{klst} = m_{kl}$ with $0 \le k \le 2m-s$ and $0 \le l \le 2n-t$, and we can formally multiply both sides of (4) by the factor $\left(\frac{1-x}{1-x^s}\right)\left(\frac{1-y}{1-y^t}\right)$, expand the denominators into geometric series, and collect equal powers of x and y on the right to get

$$\sum m_{kl} x^k y^l \equiv \sum_{\substack{k \ge 0 \\ l \ge 0}} x^k y^l \Big[\sum_{\substack{i \le k, i \equiv k \bmod s \\ j \le l, j \equiv l \bmod t}} \left(c_{ij} - c_{i,j-1} - c_{i-1,j} + c_{i-1,j-1}\right)\Big]. \tag{6}$$

Equations (5) and (6) yield a surprisingly diverse array of consequences for the original game, several of which we will now explore in the next three subsections.

4.1 When the permissible moves are fixed in size

Equation (6) applies when the moves permitted in our game are fixed in size. On the left side of (6) the indices range over $0 \leq k \leq 2m - s$ and $0 \leq l \leq 2n - t$. Consequently, a solution exists if and only if the parity conditions

$$\sum_{\substack{i \leq k,\, i \equiv k \bmod s \\ j \leq l,\, j \equiv l \bmod t}} (c_{ij} - c_{i,j-1} - c_{i-1,j} + c_{i-1,j-1}) \equiv 0 \tag{7}$$

are met for all k and l with $k > 2m - s$ or $l > 2n - t$. Furthermore, the solution must be unique, and is specified precisely by the coefficients of $x^k y^l$ in the right hand sum of (6). Because $c_{ij} = 0$ outside of the game board, this seemingly infinite set of conditions can be narrowed to those where $k \leq 2m$ and $l \leq 2n$, which are the parity conditions promised in Section 2.

When we apply the above to the checkerboard coloring (3), the parity conditions can be shown to imply that a solution exists if and only if $s|m$ and $t|n$. Moreover, in Z_2, the right side of (6) simplifies to

$$\sum_{ij} x^{2si} y^{2tj} \left[1 + x^s y^t + (1 + x^s)\left(\sum_{1 \leq k \leq t-1} y^k \right) + (1 + y^t)\left(\sum_{1 \leq l \leq s-1} x^l \right) \right]$$

where the outside sum is taken over $0 \leq i < \frac{m}{s}$ and $0 \leq j < \frac{n}{t}$. By counting monomials, we conclude that the unique winning sequence of minimum length requires exactly $\frac{2(s+t-1)}{st} \cdot mn$ moves.

4.2 When the permissible moves vary in size: the checkerboard coloring pattern

We apply equation (5) when the moves permitted can be of any size. In the case of the checkerboard coloring (3), equation (5) simplifies to

$$\begin{aligned} &\sum m_{klst} x^k y^l (1 - x^s)(1 - y^t) \equiv \\ &\equiv 1 + x^{2m} y^{2n} + (1 + x^{2m})\left(\sum_{1 \leq i \leq 2m-1} x^i \right) + (1 + y^{2n})\left(\sum_{1 \leq j \leq 2n-1} y^j \right). \end{aligned} \tag{8}$$

We can interpret this equation as a new game played on a game board of size $(2m + 1) \times (2n + 1)$. According to the right hand side, the squares on the border are colored black except for the squares at positions $(2m, 0)$ and $(0, 2n)$, which are white as well as all of the interior squares. By reinterpreting the left hand sum of (8), the moves are "corner moves" where

the colors of four squares are switched per move, namely those occupying positions $(k,l),(k,l+t),(k+s,l)$ and $(k+s,l+t)$.

Now, there is a simple solution to the original checkerboard game, already mentioned in Theorem 5, consisting of $m+n$ moves. We can easily see from our new game that this is a minimal solution, for our new coloring has exactly $4m+4m-2$ black squares, which will require at least $\lceil \frac{4m+4n-2}{4} \rceil = m+n$ corner moves.

Furthermore, we can characterize all minimal solutions. Given any solution consisting of $m+n$ corner moves, at least $m+n-2$ of these moves must account for four each of the $4m+4n-2$ black squares. The black squares at positions $(0,0)$ and $(2m,2n)$ must be divided between the last two moves; for otherwise, if one of the moves accounts for both black squares, and thus changes the colors of the squares at $(0,2n)$ and $(2m,0)$ from white to black, then the other must perform the impossible task of restoring the original colors at $(0,2n)$ and $(2m,0)$ while allowing the colors at $(0,0)$ and $(2m,2n)$ to remain unchanged. The last two moves must then account for exactly three each of the remaining black squares, and thus share a common white square. If the common square is a corner white square then each of the $m+n$ corner moves (including the last two) is composed of border squares, and in the original game, corresponds to a rectangular move that spans the board either horizontally or vertically. If the shared square is in the interior of the board, say at (k,l), then the last two moves correspond to two rectangular moves in the original game, the one with corners at $(0,0)$ and $(k-1,l-1)$, the other with corners at (k,l) and $(2m-1,2n-1)$.

4.3 When the permissible moves vary in size: arbitrary coloring patterns

One might suspect that once restrictions are lifted on the variety of coloring patterns considered and on the sizes of the rectangular moves used, the generality would preclude anything of real interest from being said. Clearly, any coloring on a $2m \times 2n$ game board can be solved and requires no more than $4mn$ moves. But can this upper bound be improved? We refer once again to equation (5), and interpret it as a new game with corner moves, as described in the previous subsection. Call the new coloring C' and let ν be the number of black squares in C'. Now clearly, a minimal solution to the new game with coloring C' must have length at least $\frac{1}{4}\nu$. On the other hand, we now describe a winning strategy for C' that requires no more than $\frac{1}{2}\nu$ moves. First, note that the multiplication of (4) by $(1-x)$ and $(1-y)$ guarantees an even number of black squares in each row and each column of C'. Divide into pairs the black squares in each row except the last. Then to each pair, apply the corner move consisting of two squares to account

for the pair and the other two squares applied to the last row. The black squares in the last row will be handled automatically. We conclude that l, the length of a minimal solution for C, has the same order of magnitude as ν, because $\frac{1}{4}\nu \leq l \leq \frac{1}{2}\nu$.

If ν is large, for instance of the order of mn, then the upper bound on l can be tightened by creating a two phase winning strategy for the new game with coloring C'. Let S be a maximal set of disjoint corner moves that each account for four black squares in coloring C'. Clearly $|S| \leq \frac{1}{4}\nu$. We play the moves in S. Next, let $C" = [c_{kl}"]$ be the new configuration consisting of the remaining black squares in C' unaccounted for by S. $C"$ inherits from C' the property that each of its rows and columns contains an even number of black squares. Furthermore, $C"$ has the delimiting property that any two of its columns must contain at most one pair of black squares occupying the same row. We use this property to bound the number of black squares in $C"$. Partition the columns of $C"$ as evenly as possible into $q = \lfloor \frac{2n+1}{\lfloor\sqrt{m}\rfloor} \rfloor$ subsets, so that each contains no more than $\lfloor\sqrt{m}\rfloor + 1$ columns. The black squares in each of the subsets can be divided between sets T_1 containing each of the black squares that shares its row with another black square in the subset, and T_2 containing the remaining black squares in the subset. By the delimiting property of $C"$ no pair of columns can contain more than one pair of black squares from T_1 in the same row, and thus $|T_1| \leq 2\binom{\lfloor\sqrt{m}\rfloor+1}{2}$. Clearly, $|T_2| \leq 2m+1$ since each row contains at most one square from T_2. Thus, an upper bound on the number of black squares in $C"$ is

$$q\left(2\binom{\lfloor\sqrt{m}\rfloor+1}{2}+2m+1\right) = \lfloor\frac{2n+1}{\lfloor\sqrt{m}\rfloor}\rfloor\left(2\binom{\lfloor\sqrt{m}\rfloor+1}{2}+2m+1\right)$$
$$\leq Kn\sqrt{m}$$

for some constant K that serves for all m and n. In fact, we can choose $K = 15$. We combine this argument with the same argument on the rows of $C"$ to find that $15\min(m\sqrt{n}, n\sqrt{m})$ is an upper bound on the number of black squares in $C"$ for all m and n. Then, applying the argument used in the previous paragraph, we win $C"$ in no more than $\frac{15}{2}\min(m\sqrt{n}, n\sqrt{m})$ additional moves. Thus, we also have the inequality $\frac{1}{4}\nu \leq l \leq \frac{1}{4}\nu + \frac{15}{2}\min(m\sqrt{n}, n\sqrt{m})$, i.e., $l = \frac{1}{4}\nu + O\big(\min(m\sqrt{n}, n\sqrt{m})\big)$. Now Theorem 7 follows. For any coloring pattern C, since $\nu \leq \min\big(2m(2n+1), 2n(2m+1)\big)$, we conclude that $l \leq mn + O\big(\min(m\sqrt{n}, n\sqrt{m})\big)$ as $m, n \to \infty$.

Finally, the inequality is in some sense best possible, for by beginning with an entirely black pattern for C' except for the last row and column, and tracing back to the corresponding coloring for C, one can find that the $2m \times 2n$ game beginning with coloring $C = [c_{ij}]$ defined by $c_{ij} = 1$ if

and only if i and j are both even, requires exactly mn moves for a minimal solution.

5 Conclusion

In this section we state some results that highlight the versatility of the techniques presented in this paper, and suggest further directions that the reader may wish to pursue. Theorems 9 and 11 follow directly by the techniques used in Sections 4.3 and 4.2, respectively, and are left for the reader as exercises. Roughly speaking, for board colorings generated by tiles, the former one bridges the gap between Theorems 5 and 7 by the degree of complexity imposed on the tiles. In fact, Theorem 9 extends Theorem 5 to colorings defined by simple tiles, and it refines Theorem 7 by allowing complex tiles. Theorem 10 relies on a more extensive argument rooted in Section 4.1. We will present the proof in a forthcoming paper.

Theorem 9 *Let $M = [m_{ij}]$ be a $(0,1)$ matrix of size $s \times t$. For any m and n with $s|m$ and $t|n$ let C_{mn} be a $2m \times 2n$ matrix consisting of $\frac{2m}{s} \cdot \frac{2n}{t}$ block copies of matrix M, and interpret C_{mn} as a coloring configuration in a game of size $2m \times 2n$, where the moves are arbitrarily sized rectangular patterns.*

(i) If the matrix M has the property that each row of M is either equal to or the binary complement of the first row, then the number of moves in a minimal solution for C_{mn} is asymptotic to $\frac{c}{s} \cdot m + \frac{d}{t} \cdot n$ as $m, n \to \infty$, where

$$c = |\{i \mid m_{i,0} + m_{i+1,0} \equiv 1 \bmod 2,\ 0 \le i < s\}|$$

and

$$d = |\{j \mid m_{0,j} + m_{0,j+1} \equiv 1 \bmod 2,\ 0 \le j < t\}|$$

are the number of color switches in all rows and columns of M, respectively.

(ii) Otherwise, the number of moves in a minimal solution for C_{mn} is asymptotic to $\frac{k}{st} \cdot mn$ as $m, n \to \infty$, where

$$k = |\{(i,j) \mid m_{ij} + m_{i,j+1} + m_{i+1,j} + m_{i+1,j+1} \equiv 1 \bmod 2,\ 0 \le i < s,\ 0 \le j < t\}|.$$

Here we let $m_{sj} = m_{0j}, m_{it} = m_{i0}$ and $m_{st} = m_{00}$.

Theorem 10 *Suppose the board is an arbitrarily colored circular one with n squares, and the moves are arcs of adjacent squares of length r. Let $g = \gcd(n, r)$ and $S_a = \sum_{i \equiv a \bmod g} c_i$, for all $0 \le a < g$. If $\frac{r}{g}$ is even then the game can be won if and only if $S_a \equiv 0 \bmod 2$ for all a. On the other hand, if $\frac{r}{g}$ is odd then it can be won if and only if all S_a, $0 \le a < g$, have the same parity.*

Theorem 11 *If the board is a $2n \times 2n$ checkerboard and the moves are arbitrarily sized squares then a minimal solution has length $4n-2$ and can be easily described.*

References

[1] E. R. Berlekamp, J. H. Conway, and R. K. Guy, *Winning Ways for Your Mathematical Plays*, vol. 2, Academic Press, London–New York, 1982

[2] F. Galvin, Solution to Problem 88-8, *Math. Intelligencer* **11** (1989), 31–32.

[3] R. Honsberger, *In Pólya's Footsteps*, Mathematical Association of America, 1997

[4] O.P. Lossers, An all-ones problem, *Amer. Math. Monthly* **100** (1993), 806–807.

[5] L. Lovász, *Combinatorial Problems and Exercises*, North-Holland, New York, 1979

[6] D. H. Pelletier, Merlin's magic square, *Amer. Math. Monthly* **94** (1987), 143–150.

[7] D. L. Stock, Merlin's magic square revisited, *Amer. Math. Monthly* **96** (1989), 608–610.

[8] K. Sutner, Linear cellular automata and the Garden-of-Eden, *Math. Intelligencer* **11** (1989), 49–53.

[9] P. Zeitz, *The Art and Craft of Problem Solving*, Wiley, 1999

1	0	0
1	1	0
0	0	1

Merlin's Magic Squares

Liz Arnold and Tiffany Blaszak

12-09-02

Abstract

The purpose of this paper is to solve the game Merlin's Magic Squares by using our knowledge of Linear Algebra.

Introduction

Merlin's Magic Squares is a hand held electronic game made by Parker Brothers. It consists of a 3x3 array with nine boxes, each containing either a one or a zero. For our purposes we will be using the online version of the game located at http://www.cut-the-knot.com/ctk/Merlin.shtml. When you start the game, the player is given an initial configuration. The player must then turn this initial configuration into the final configuration. When the player reaches this final configuration, the player has won the game. Figure 1 shows what an initial configuration could look like at the start of the game.

In this game, when the player presses any one of the nine buttons, a certain subset of the buttons is altered according to the rules of the game. Figures 2(a), 2(b) and 2(c) show what happens when you press a corner button, a side button and the middle

1	0	0
1	1	0
0	0	1

Figure 1: Initial Configuration.

0	1	0
0	0	0
0	0	1

(a) Corner.

0	1	1
0	0	1
0	0	0

(b) Side.

0	0	1
1	1	0
0	1	0

(c) Middle.

Figure 2: Effect of pressing a button.

1	0	0
1	1	0
0	0	1

1	2	3
4	5	6
7	8	9

Figure 3:

button respectively. This same pattern occurs for every corner button and side button that the player presses.

The nine buttons on Merlin's Magic Squares are numbered one through nine as shown in Figure 3.

Table 1 describes the effects of pressing each button.

Good Stuff

Notice that when the player pushes the same button an odd number of times, it will have the same effect as if we only pushed the button once! Here, we will show what happens to the initial configuration when the same button is pushed once, twice and three times. Notice how the effect of pushing the button once and three times produces the same configuration. Figures 4(b) and 4(d) show the effects of hitting the button once and three times respectively. Since pushing a button an odd number of times will have the same affect as if we only pushed the button once, Figure 4(b) and 4(d) will

1	0	0
1	1	0
0	0	1

Pressing button	1,	alters the state of buttons	1, 2, 4 and 5
	2,		1, 2, and 3
	3,		2, 3, 5 and 6
	4,		1, 4 and 7
	5,		2, 4, 5, 6 and 8
	6,		3, 6 and 9
	7,		4, 5, 7 and 8
	8,		7, 8 and 9
	9		5, 6, 8 and 9

Table 1:

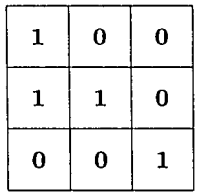
have the same configuration.

Notice in Figure 5(c) that the pressing a button twice produces the same configuration as not pressing it at all.

Figures 6 and 7 demonstrate that the order in which the player presses the buttons does not affect the final configuration.

The Goal

The goal of Merlin's Magic Squares is to reach this final configuration as shown in Figure 8.



1	0	0
1	1	0
0	0	1

(a) Initial Configuration.

0	1	0
0	0	0
0	0	1

(b) First press.

1	0	0
1	1	0
0	0	1

(c) Second press.

0	1	0
0	0	0
0	0	1

(d) Third press.

Figure 4: Effect of pressing the button in the upper left hand corner three times.

1	0	0
1	1	0
0	0	1

(a) Initial Configuration.

0	1	0
0	0	0
0	0	1

(b) First press.

1	0	0
1	1	0
0	0	1

(c) Second press.

Figure 5: Effect of pressing the button in the upper left hand corner twice.

0	1	0
0	0	0
0	0	1

(a) Button 1.

0	1	1
0	0	1
0	0	0

(b) Button 6.

0	0	1
1	1	0
0	1	0

(c) Button 5.

Figure 6: The effect of pressing buttons 1 then 6 then 5.

1	0	0
1	1	0
0	0	1

1	0	1
1	1	1
0	0	0

(a) Button 6.

1	1	1
0	0	0
0	1	0

(b) Button 5.

0	0	1
1	1	0
0	1	0

(c) Button 1.

Figure 7: The effect of pressing the same buttons in a different order (6, 5, 1).

1	1	1
1	0	1
1	1	1

Figure 8: Winning Configuration.

1	0	0
1	1	0
0	0	1

1	0	0
1	1	0
0	0	1

Using Linear Algebra

We will be using the theory of vector spaces to show a mathematical model of the game. For this game we will be working in the binary field in which the only elements are 1 and 0, and where the field operations are addition and multiplication in modulo 2 shown in the following table:

Addition

+	0	1
0	0	1
1	1	0

Multiplication

×	0	1
0	0	0
1	0	1

For this exercise, we can represent the configuration of our array as a column vector with nine components. It will be helpful to number its boxes and representative vector in the manner shown below.

1	**2**	**3**
4	**5**	**6**
7	**8**	**9**

$$\mathbf{v} = \begin{pmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \end{pmatrix}$$

We'll call our initial configuration $\mathbf{v}_p$ and our winning configuration $\mathbf{v}_w$.

1	0	0
1	1	0
0	0	1

$$\mathbf{v}_p = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} \quad \text{and} \quad \mathbf{v}_w = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 0 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

The game is designed such that for every possible button strike, there is a representative vector $\mathbf{u}_i$ that is added to the original vector $\mathbf{v}_p$ to produce a new vector $\mathbf{v}_{p'}$.

$$\mathbf{v}_p + \mathbf{u}_i = \mathbf{v}_{p'}$$

Example 1 *Pressing button 1 adds* $\mathbf{u}_1$ *to our initial vector as follows:*

$$\mathbf{v}_p + \mathbf{u}_1 = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} + \begin{pmatrix} 1 \\ 1 \\ 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} = \mathbf{v}_p'$$

Note that ones appear in $\mathbf{u}_1$ corresponding to the boxes whose states are changed while zeros represent no change.

1	0	0
1	1	0
0	0	1

All possible button strikes then, can be represented by 9 different u_i's.

$$\mathbf{u}_1 = \begin{pmatrix}1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0\end{pmatrix}^T$$

$$\mathbf{u}_2 = \begin{pmatrix}1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0\end{pmatrix}^T$$

$$\mathbf{u}_3 = \begin{pmatrix}0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0\end{pmatrix}^T$$

$$\mathbf{u}_4 = \begin{pmatrix}1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0\end{pmatrix}^T$$

$$\mathbf{u}_5 = \begin{pmatrix}0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 0\end{pmatrix}^T$$

$$\mathbf{u}_6 = \begin{pmatrix}0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1\end{pmatrix}^T$$

$$\mathbf{u}_7 = \begin{pmatrix}0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0\end{pmatrix}^T$$

$$\mathbf{u}_8 = \begin{pmatrix}0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1\end{pmatrix}^T$$

$$\mathbf{u}_9 = \begin{pmatrix}0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1\end{pmatrix}^T$$

With this information we can now express the ability to reach the winning configuration, $\mathbf{v}_w$, by our ability to write it as a linear combination of $\mathbf{v}_p$ and our $\mathbf{u}_i$'s.

$$\mathbf{v}_w = \mathbf{v}_p + s_1\mathbf{u}_1 + s_2\mathbf{u}_2 + \cdots + s_9\mathbf{u}_9$$

Our only unknowns at this point are our scalars s_i where

$$s_i = \begin{cases} 1, & \text{if button } \#i \text{ is pressed,} \\ 0, & \text{otherwise.} \end{cases}$$

Now we rearrange our terms.

$$\mathbf{v}_w - \mathbf{v}_p = s_1\mathbf{u}_1 + s_2\mathbf{u}_2 + \cdots + s_9\mathbf{u}_9$$

1	0	0
1	1	0
0	0	1

Since we are performing addition in modulo 2,

$$\mathbf{v}_w - \mathbf{v}_p = \mathbf{v}_w + \mathbf{v}_p$$

So,

$$\mathbf{v}_w + \mathbf{v}_p = s_1\mathbf{u}_1 + s_2\mathbf{u}_2 + \cdots + s_9\mathbf{u}_9$$

Now we'll place our $\mathbf{u}_i$'s into the columns of a 9×9 matrix A.

$$A = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix}$$

Since our scalars multiply the columns of A, we can represent this multiplication as

$$Ax = b,$$

or for our purposes,

$$A\mathbf{s} = \mathbf{v}_w + \mathbf{v}_p.$$

The sequence of keystrokes needed to win the game is found in $\mathbf{s}$, which is the solution to the system $A\mathbf{s} = \mathbf{v}_w + \mathbf{v}_p$, which we can rewrite to find s.

$$\begin{aligned} A\mathbf{s} &= \mathbf{v}_w + \mathbf{v}_p \\ A^{-1}\mathbf{s} &= A^{-1}(\mathbf{v}_w + \mathbf{v}_p) \\ \mathbf{s} &= A^{-1}(\mathbf{v}_w + \mathbf{v}_p) \end{aligned}$$

1	0	0
1	1	0
0	0	1

Now we can easily solve for s using A^{-1}.

We will need to invert A, but we must first determine whether A is invertible or not by computing the determinant.

$$\text{If } \det(A) \neq 0, \text{ then } A^{-1} \text{ exists.}$$

To find the determinant of A we will row reduce the matrix in modulo 2 and then multiply the diagonal elements:

Adding $row1$ to $row2$ gives us

$$\begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix}.$$

Switching $row2$ with $row3$ produces a pivot in $row2 : column\ 2$ of A.

$$\begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix}$$

Continuing in this manner, we obtain an upper triangular matrix U.

$$U = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Multiplying the diagonal elements, you can see that the determinant is 1, so the matrix is indeed invertible.

A formula for calculating A^{-1} is:

$$A^{-1} = \frac{1}{|A|} C^T$$

where C is the cofactor matrix of A.

$$C = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 1 \end{pmatrix}$$

1	0	0
1	1	0
0	0	1

1	0	0
1	1	0
0	0	1

Here, we compute C using the Matlab's command C=mod(cofactor(A),2), which finds the cofactor matrix operating in modulo 2.

Since $|A| = 1$, we know that $A^{-1} = C^T$ so,

$$A^{-1} = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 1 \end{pmatrix}.$$

Solving for $\mathbf{s}$ will show us the buttons we need to press to win the game. Recall that

$$A^{-1}(\mathbf{v}_w + \mathbf{v}_p) = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 1 \\ 1 \\ 1 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 1 \\ 0 \\ 1 \end{pmatrix} = \mathbf{s}.$$

The ones in $\mathbf{s}$ tell us that pressing buttons 5, 6, 7, and 9 will give us our winning configuration. The winning play is shown in Figure 9.

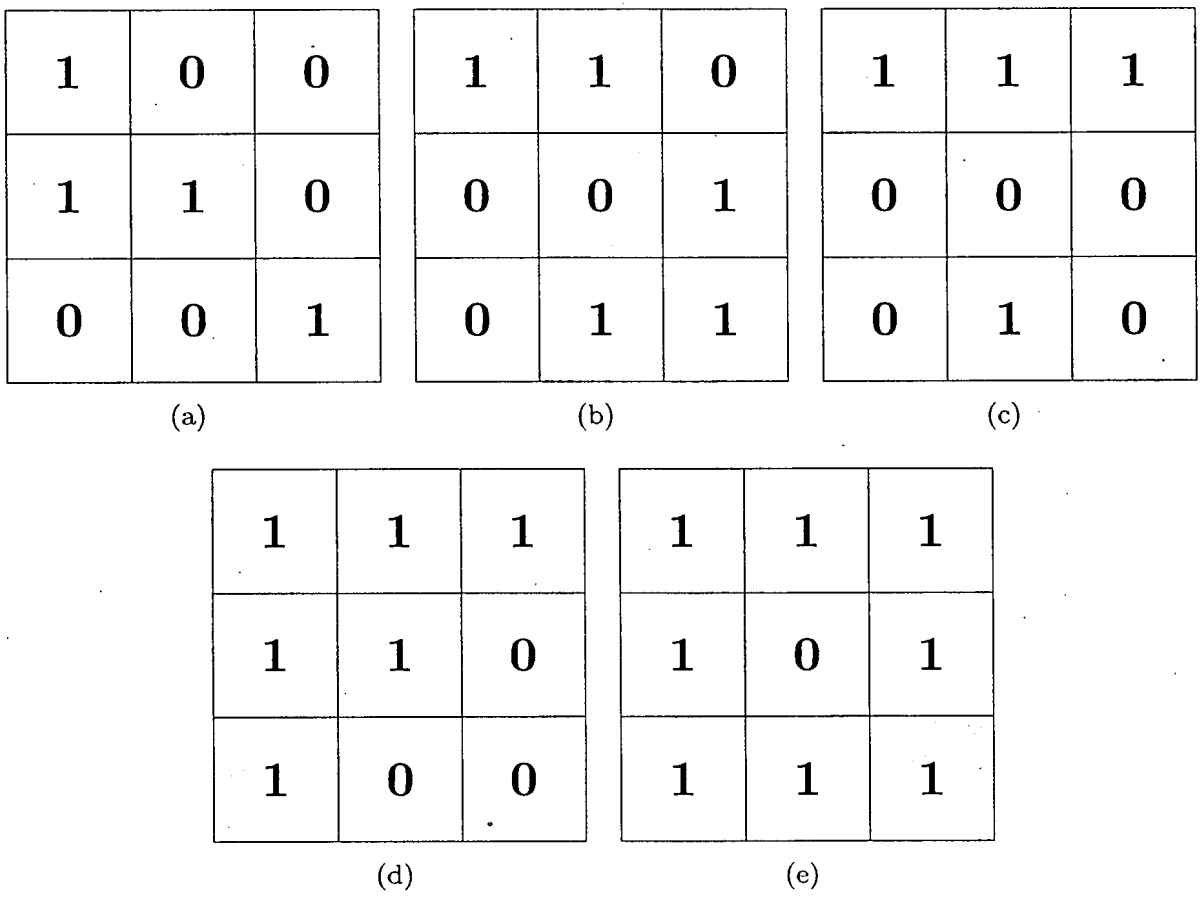


Figure 9: Pressing buttons 5, 6, 7, and 9 wins the game.

1 0 0
1 1 0
0 0 1

1	0	0
1	1	0
0	0	1

Conclusion

Now we have a surefire method for winning Merlin's Magic Squares given any initial configuration. In fact, it is possible to achieve any desired configuration by simply replacing our original goal configuration with some new configuration of our choice, then following the same steps. This application, though not particularly technical or complicated, demonstrates that we can apply linear algebra to real situations. Our simple example shows us a fun way to exercise our linear algebra skills.

References

[1] Arnold, David. *College of the Redwoods*

[2] Pelletier, Don. *Merlin's Magic Square.* American Mathematical Monthly, volume 94, Issue 2(Feb, 1987), 143-150

[3] Strang, Gilbert *Introduction to Linear Algebra*

[4] Bogomolny, Alex *Merlin's Magic Squares*

http://www.cut-the-knot.com/ctk/Merlin.shtml